Title: AUTOMATICALLY INSERTING INSTRUCTIONS INTO A PROGRAM THAT CAUSE A
THREAD CONTEXT SWAP (As Amended)
Inventor: Johnson et al.   Attorney Docket No.: 42P16856   Filing Date: December 12, 2003
Filed EFS-WEB on July 9, 2007

1/13   Replacement Sheet

FIG. 1B

FIG. 1A

Title: AUTOMATICALLY INSERTING INSTRUCTIONS INTO A PROGRAM THAT CAUSE A
THREAD CONTEXT SWAP (As Amended)
Inventor: Johnson et al.   Attorney Docket No.: 42P16856   Filing Date: December 12, 2003
Filed EFS-WEB on July 9, 2007

2/13   Replacement Sheet

FIG. 2

Title: AUTOMATICALLY INSERTING INSTRUCTIONS INTO A PROGRAM THAT CAUSE A
THREAD CONTEXT SWAP (As Amended)

Inventor: Johnson et al.   Attorney Docket No.: 42P16856   Filing Date: December 12, 2003
Filed EFS-WEB on July 9. 2007

3/13   Replacement Sheet

FIG. 3A

Title: AUTOMATICALLY INSERTING INSTRUCTIONS INTO A PROGRAM THAT CAUSE A
THREAD CONTEXT SWAP (As Amended)
Inventor: Johnson et al.   Attorney Docket No.: 42P16856   Filing Date: December 12, 2003
Filed EFS-WEB on July 9, 2007

4/13   Replacement Sheet

FIG. 3B

Title: AUTOMATICALLY INSERTING INSTRUCTIONS INTO A PROGRAM THAT CAUSE A
THREAD CONTEXT SWAP (As Amended)

Inventor: Johnson et al.   Attorney Docket No.: 42P16856   Filing Date: December 12, 2003
Filed EFS-WEB on July 9, 2007

5/13   **Replacement Sheet**

existing: *
end block: *
local block: 3
start block: 7
compute #: 10

3 compute | relinquish | 7 compute

200

existing: 7
end block: 2
local block: 10
start block: 1
compute #: 13

2 compute | relinquish | 10 compute | relinquish | 1 compute

204

existing: 7
end block: 0
local block: 0
start block: 0
compute #: 6

6 compute

202

existing: 13, 1
end block: 10
local block: 2
start block: *
compute #: 12

10 compute | relinquish | 2 compute

206

FIG. 3C

Title: AUTOMATICALLY INSERTING INSTRUCTIONS INTO A PROGRAM THAT CAUSE A THREAD CONTEXT SWAP (As Amended)

Inventor: Johnson et al.   Attorney Docket No.: 42P16856   Filing Date: December 12, 2003
Filed EFS-WEB on July 9, 2007

6/13   **Replacement Sheet**

200:
existing: *
end block: *
local block: 3,5
start block: 2
compute #: 10

3 compute
relinquish
5 compute
relinquish
2 compute

204:
existing: 2
end block: 2
local block: 5,5
start block: 1
compute #: 13

2 compute
relinquish
5 compute
relinquish
5 compute
relinquish
1 compute

202:
existing: 2
end block: 3
local block: 0
start block: 3
compute #: 6

3 compute
relinquish
3 compute

206:
existing: 3, 1
end block: 1
local block: 5,4,2
start block: *
compute #: 12

1 compute
relinquish
5 compute
relinquish
4 compute
relinquish
2 compute

FIG. 3D

Title: AUTOMATICALLY INSERTING INSTRUCTIONS INTO A PROGRAM THAT CAUSE A
THREAD CONTEXT SWAP (As Amended)

Inventor: Johnson et al.   Attorney Docket No.: 42P16856   Filing Date: December 12, 2003
Filed EFS-WEB on July 9, 2007
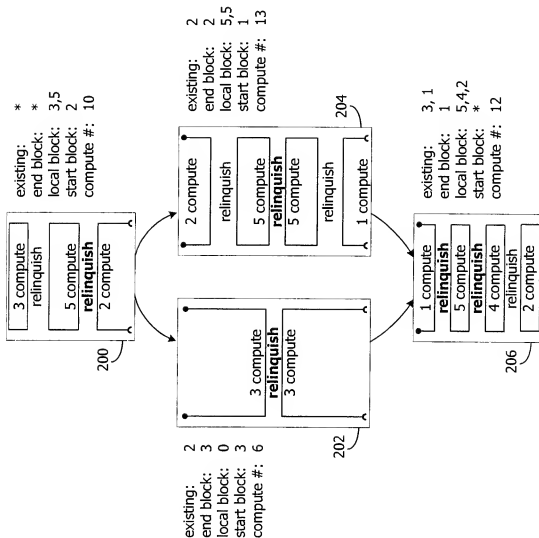
7/13   Replacement Sheet

// for wholly included compute blocks

for each compute block wholly contained in node

if block_size > threshold

number_blocks = ceiling(block_size, threshold)

insert relinquish instructions to break up block into ~ equal
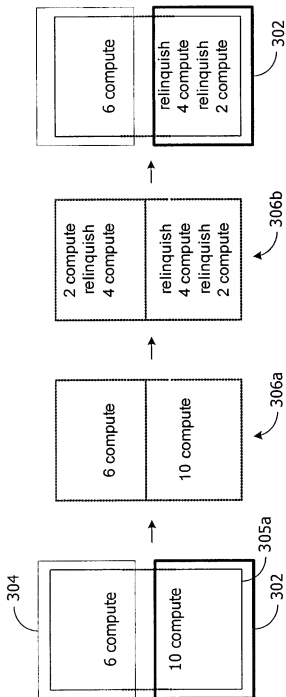number_blocks

300

FIG. 4A

Title: AUTOMATICALLY INSERTING INSTRUCTIONS INTO A PROGRAM THAT CAUSE A
THREAD CONTEXT SWAP (As Amended)

Inventor: Johnson et al.   Attorney Docket No.: 42P16856   Filing Date: December 12, 2003
                            Filed EFS-WEB on July 9, 2007

8/13   Replacement Sheet

FIG. 4B

Title: AUTOMATICALLY INSERTING INSTRUCTIONS INTO A PROGRAM THAT CAUSE A
THREAD CONTEXT SWAP (As Amended)

Inventor: Johnson et al.   Attorney Docket No.: 42P16856   Filing Date: December 12, 2003
Filed EFS-WEB on July 9, 2007

9/13   Replacement Sheet

// blocks started in ancestor and terminated in current node

if (min (ancestor start block) + end_block) < threshold goto exit

308
number_blocks = ceiling((min(ancestor start block)+ end_block) / threshold
new_size = (min(ancestor start block) + end_block) / number_blocks
instruction_number = min(ancestor start block) modulo new_size

if (instruction_number > end_block) goto exit
end_block = instruction_number

310
insert relinquish instructions, starting at instruction_number,
every (new_size + 1) instructions

FIG. 4C

Title: AUTOMATICALLY INSERTING INSTRUCTIONS INTO A PROGRAM THAT CAUSE A
THREAD CONTEXT SWAP (As Amended)

Inventor: Johnson et al.   Attorney Docket No.: 42P16856   Filing Date: December 12, 2003
Filed EFS-WEB on July 9, 2007

10/13   Replacement Sheet

// blocks started in this node and terminated in descendent

if (start_block + min (descendent end block)) < threshold skip this processing

// Determine where to insert the first relinquish instruction

312   number_blocks = ceiling((start_block + min(descendent end block) / threshold)
new_size = (start_block + min (descendent end block) / number_blocks
instruction_number = min (descendent end block) modulo new_size

314   insert relinquish instructions, starting instruction_number
from the end of the node, every (new_size + 1) instructions

FIG. 4D

Title: AUTOMATICALLY INSERTING INSTRUCTIONS INTO A PROGRAM THAT CAUSE A
THREAD CONTEXT SWAP (As Amended)

Inventor: Johnson et al.   Attorney Docket No.: 42P16856   Filing Date: December 12, 2003
Filed EFS-WEB on July 9, 2007

11/13   **Replacement Sheet**

// for nodes having no relinquish instructions

316   // Determine size of smallest contiguous block of instructions
     size = min (descendent end block) + compute_count + min (ancestor start block)
     if size < threshold goto exit

// Compute where to insert first relinquish instruction in this node
     number_blocks = ceiling(size / threshold)
     new_size = size / number_blocks
318   instruction_number = min (ancestor start block) modulo new_size
     if instruction_number > compute_count goto exit

     insert relinquish instructions, starting at
     instruction_number, every (new_size + 1) instructions

FIG. 4E

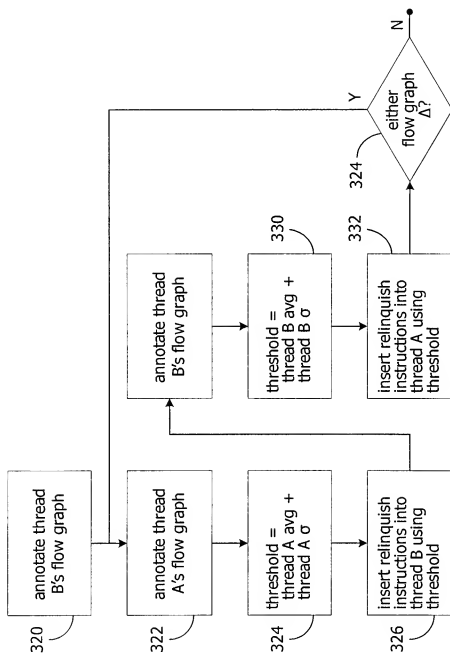Title: AUTOMATICALLY INSERTING INSTRUCTIONS INTO A PROGRAM THAT CAUSE A
THREAD CONTEXT SWAP (As Amended)

Inventor: Johnson et al.   Attorney Docket No.: 42P16856   Filing Date: December 12, 2003
                              Filed EFS-WEB on July 9, 2007

12/13   Replacement Sheet

annotate thread B's flow graph — 320

annotate thread A's flow graph — 322

threshold =
thread A avg +
thread A σ — 324

insert relinquish instructions into thread B using threshold — 326

annotate thread B's flow graph

threshold =
thread B avg +
thread B σ — 330

insert relinquish instructions into thread A using threshold — 332

either flow graph Δ? — 324

Y

N

FIG. 5

Title: AUTOMATICALLY INSERTING INSTRUCTIONS INTO A PROGRAM THAT CAUSE A
THREAD CONTEXT SWAP (As Amended)

Inventor: Johnson et al.   Attorney Docket No.: 42P16856   Filing Date: December 12, 2003
                             Filed EFS-WEB on July 9, 2007

13/13   **Replacement Sheet**

FIG. 6